Background subtraction driven seeds selection for moving objects segmentation and matting

Bineng Zhonga,d, Yan Chen a, Yewang Chen a, Rongrong Jib, Ying Chen c, Duansheng Chen a, Hanzi Wangd

a Department of Computer Science and Engineering, Huaqiao University, Jimei Road, No.668, Xiamen, Fujian, China
b Department of Electronic Engineering, Columbia University, NY, USA
c Department of Basic Sciences, Beijing Electronic Science and Technology Institute, Beijing, China
d School of Information Science and Technology, Xiamen University, Xiamen, Fujian, China

ARTICLE INFO
Article history:
Received 20 April 2012
Received in revised form 4 August 2012
Accepted 6 October 2012
Communicated by: Ran He
Available online 26 October 2012

Keywords:
Segmentation
Matting
Background subtraction
Moving object detection
Dynamic scenes
Neighboring Image Patches Embedding

ABSTRACT
In this paper, we address the difficult task of moving objects segmentation and matting in dynamic scenes. Toward this end, we propose a new automatic way to integrate a background subtraction (BGS) and an alpha matting technique via a heuristic seeds selection scheme. Specifically, our method can be divided into three main steps. First, we use a novel BGS method as attention mechanisms, generating many possible foreground pixels by tuning it for low false-positives and false-negatives as much as possible. Second, a connected components algorithm is used to give the bounding boxes of the labeled foreground pixels. Finally, matting of the object associated to a given bounding box is performed using a heuristic seeds selection scheme. This matting task is guided by top-down knowledge. Experimental results demonstrate the efficiency and effectiveness of our method.

1. Introduction

Accurate moving objects detection and matting in videos has long been a basic problem in computer vision that attracts much research interest due to its wide variety of applications, including intelligence video surveillance [36–38,42], image editing [39], human machine interfaces, special effects in motion pictures, and so on. Despite much progress has been made in the last two decades, the problem has remained challenging to this date. Challenges in the problems include dynamic background (e.g., water ripples, swaying trees and rain), illumination changes, tremendous amount of manual interaction to provide trimap seeds, etc.

To achieve robust moving objects detection even in dynamic scenes, a number of background subtraction (BGS) methods have been proposed. BGS method can be classified into three categories: (1) the pixel-based methods that use only the pixel color or intensity information to make the decision; (2) the patch-based methods that consider some statistics of neighborhoods; (3) the hybrid methods that combine the pixel-based and the patch-based BGS methods.

1.1. Pixel-based methods

One popular technique is to model each pixel's color in a video frame with a Gaussian distribution [1]. This model does not work well in the case of dynamic scenes. To deal with this problem, Gaussian Mixture Model (GMM) [2] is used to model each pixel. But it cannot adapt to the case where the background has quick variations [3]. Numerous improvements of the original method developed by Stauffer and Grimson [2] have been proposed over the recent years and a good survey of these improvements is presented in [4]. In the W4 system [5], the background scene is statically modeled by the minimum and maximum intensity values and maximal temporal derivative for each pixel recorded over some period. A non-statistical clustering technique to construct a background model is presented in [6]. The background is encoded on a pixel-by-pixel basis and samples at each pixel are clustered into a set of codewords.

1.2. Patch-based methods

Elgammal et al. [7] are among the first to utilize the kernel density estimation (KDE) technique to model the background's color distribution, which has been successfully applied in the BGS literature. Another significant contribution of this work is the
incorporation of the spatial constraints into the formulation of foreground classification. In the second phase of their approach, the pixel values that could be explained away by distributions of neighboring pixels are reclassified as background, allowing for greater resilience against dynamic backgrounds. In [8], the background and foreground models are first constructed via a KDE technique separately, which are then used competitively in a MAP-MRF decision framework. Mittal and Paragios [9] propose the use of variable bandwidths for KDE to enable modeling of arbitrary shapes of the underlying density in a more natural way. Parag and Elgammal [10] use a boosting method (RealBoost) to choose the best feature to distinguish the foreground for each of the areas in the scene. However, one key problem with the KDE techniques is their high computational requirement due to the large number of samples needed to model the background. A Bayesian framework that incorporates spectral, spatial, and temporal features to characterize the background appearance is proposed in [11]. Under this framework, the background is represented by the most significant and frequent features, i.e., the principal features, at each pixel. Seki et al. [12] propose a BGS method based on the co-occurrence of image variations, which can be regarded as narrowing the background image variations by estimating the background image pattern in each image block from the neighboring image patterns in the input image. Some authors model the background using texture features. Heikilä and Pietikäinen [13] propose an approach based on the discriminative LBP histogram. However, simple grayscale operations make LBP rather sensitive to noise and it is also not so efficient on uniform regions. Yao and Odobez [14] propose a multiple layer background model which makes use of the LBP texture feature and color feature. In [15], the background is firstly divided into three types of regions—flat, sketchable and textured region according to a primal sketch representation. Then, the three types of regions are modeled respectively by Mixture of Gaussians, image primitives and LBP histograms. Finally, the geometry information obtained from camera calibrations is used to further reduce the false alarms. Ko et al. [16] have developed a BGS scheme that analyzes the temporal variation of intensity or color distributions, instead of either looking at temporal variation of point statistics, or the spatial variation of region statistics in isolation. Dalley et al. [17] introduce a new image generation model that takes into account the spatial uncertainty of dynamic background textures. In their model, they allow the pixels to be generated from nearby Gaussians. Mahadevan and Vasconcelos [18] view BGS as a problem of saliency detection: the background points are those considered not salient by suitable comparison of object and background appearance and dynamics. Zhong et al. [19] propose a novel feature extraction method, Neighboring Image Patches Embedding (NIPE), for background modeling. The NIPE feature vector, whose components are similarities between current image patch and its neighbors, describes mainly the mutual relationship between neighboring patches.

1.3. Hybrid methods

In [3,20], a three-stage algorithm is separately presented, which operates respectively at pixel, region and frame level. Tian et al. [21] first use BGS to get a set of candidate foreground pixels, then use foreground analysis to remove the false alarm pixels of the detected foreground regions. In [22], a scene is coarsely represented as the union of pixel layers and the foreground objects are detected by propagating these layers using a maximum-likelihood assignment. However, the limitations of the method are high-computational complexity and the requirement of an extra offline training step. In [32], the background is modeled as a set of warping layers, where at any given time, different layers may be visible due to the motion of an occluding layer. Since low-rank subspaces have been a powerful tool in image processing and machine learning [40], Candes et al. [41] propose a robust PCA (principal component analysis) method for background subtraction. However, the SVD (singular value decomposition) used to perform the robust PCA is too slow for the real-time applications.

In a word, the pixel-based methods, though quite successful, can be hindered by their lack of explicit encoding of statistics of neighborhoods—one might, for example, generate many false-positives (i.e., false foreground pixels) in dynamic scenes. The patch-based methods, though insensitive to the noises and the small movement of the dynamic scene, may lead to distortion at the boundary of moving objects. Although the hybrid methods can further improve the detection results by exploiting the complementary strengths of pixel-based and patch-based BGS methods, accurately extracting moving objects in dynamic scenes remains a very challenging problem.

To accurately extracting moving objects in a video, a number of authors have viewed the problem as a video matting problem, in which a high-quality alpha matte and foreground are pulled from a video sequence. However, most of the techniques require a known background (e.g., a blue screen [23,33]), several key frames [24], or tremendous amount of manual interaction to provide trimap seeds [25,26,34] and scribbles [31,35]. Please refer to [27] for a more complete image and video matting survey.

The following question naturally arises: Can we unite the strengths of a BGS and an alpha matting technique to automate the process of accurate moving objects detection and matting in dynamic scenes? Our answer is yes. In this paper, we propose a new automatic matting way to extract a high-quality alpha matte and moving objects from the dynamic scenes driven by BGS-based seeds selection. First, we use a novel BGS method as attention mechanism, generating many possible foreground pixels by tuning it for low false-positives and false-negatives as much as possible. To achieve the two goals, we adopt the following strategy: (1) We design a novel feature extraction framework, Neighboring Image Patches Embedding (NIPE) for robust and efficient moving objects detection with low false-positives. (2) As the NIPE feature vector is a patch-based feature, the NIPE-based BGS method may lead to high false-negatives at the boundary of moving objects. We further exploit the complementary strengths of the pixel-based and the patch-based BGS methods via a foreground AND operator to get the low false-negatives. Second, a connected components algorithm is used to give the bounding boxes of the labeled foreground pixels. Finally, at each extracted bounding box, we compute local trimap matting via a heuristic seeds selection scheme, in which the labeled foreground pixels in the bounding box are used as the foreground seeds and the pixels in a window of slightly larger extent than the bounding box (that) are used as the background seeds. Experimental results show that this strategy leads to accurate moving objects detection and matting even in dynamic scenes.

The rest of the paper is organized as follows: Section 2 introduces the overview of our method. Then, we describe the framework of NIPE, verify its efficacy for background modeling, and discuss the construction and maintenance of the NIPE-based background modeling in Section 3. In Section 4, we briefly introduce the GMM-based background modeling method. The detailed algorithm of heuristic seeds selection for matting is described in Section 5. Experimental results are given in Section 6, and we conclude in Section 7.

2. Overview of our method

To accurately extract moving objects in dynamic scenes, we combine a background subtraction and an alpha matting
technique via a heuristic seeds selection scheme. The flowchart of our method is shown in Fig. 1, in which the complementary pixel color information and statistics of neighborhoods are first exploited to generating candidate BGS maps. Then, an automatic matting technique driven by a heuristic seeds selection scheme is utilized to extract a high-quality alpha matte and moving objects from the dynamic scenes.

Specifically, as shown in Fig. 2, we first run the NIPE-based and the GMM-based background subtraction method that output a background subtraction map respectively. Second, these background subtraction maps are combined via a foreground AND operator as attention mechanism to finally get a more robust and accurate background subtraction map. This is a natural fusion because the NIPE-based method using block feature and the GMM-based method using pixel feature have complementary strengths, in which the NIPE-based method is insensitive to the noises and the small movement of dynamic scenes and the GMM-based method contains rich information. Third, we use a connected components algorithm to give the green bounding boxes of the labeled foreground pixels. Finally, at each extracted green bounding box, we compute local trimap matting via a heuristic seeds selection scheme, in which the labeled foreground pixels in the green bounding box are used as the foreground seeds and the pixels outside the red bounding boxes and inside the yellow bounding boxes are used as the background seeds. The yellow bounding boxes are obtained by doubling the size of the green bounding boxes, and the red bounding boxes are obtained by extending the size of the green bounding boxes in each dimension a quarter. This matting task is guided by top-down knowledge.

Below we give a detailed description about each component of our method.

3. Neighboring Image Patches Embedding (NIPE) for BGS

3.1. Basic description

In this section, we describe the novel feature extraction framework of NIPE. We partition an image into a set of non-overlapping patches with $N \times N$ pixels and represent each patch by a $K$ dimensional NIPE vector whose components are similarities between current image patch and its $K$ neighbors. The similarities could be computed using various robust image properties, such as matching of texture based descriptors (e.g., LBP [13]), distribution based descriptors (e.g., SIFT [28]), differential descriptors (e.g., local derivatives [29]), and so on.

Please see Fig. 3 for representing an image patch $P_t$ at time $t$ as a NIPE vector $V_{P_t}$ whose components are similarities between this image patch and its $K$ neighbors. The image patch $P_t$ is compared with surrounding $K$-neighbor patches centered at $P_t$, using various robust image properties. We then get a $K$ dimensional vector $V_{P_t} = d_{P_t1}, d_{P_t2}, \ldots, d_{P_tK}$ to describe the image patch $P_t$, where $d_{P_tk}$ is computed between the patch $P_t$ and one of its $K$-neighbor patches. We call this vector $V_{P_t}$ a NIPE vector. Therefore, we can effectively use the much lower dimensional NIPE vector to represent a high

![Fig. 1. Flowchart of our method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)](image1)

![Fig. 2. Illustration of the heuristic seeds selection scheme for moving objects matting and segmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)](image2)
dimensional image patch (typically $K \times N \times N$), and construct and update the background model with the NIPE vectors. Every patch in the image at time $t$ can be modeled using NIPE in the similar way as described above.

NIPE vector has several properties that favor its usage in background modeling. Since NIPE vector is obtained by computing the similarities between neighboring patches, our method can tolerate dynamic background and monotonic gray-scale changes in natural scenes. A low dimensional NIPE vector representation avoids expensive computation, which is an important property for practical real-time applications. Furthermore, NIPE belongs to nonparametric methods, which means that no assumptions about the underlying distributions are needed.

### 3.2. Efficacy of NIPE for background modeling

In this section, we illustrate the efficacy of NIPE for dynamic background modeling by using an image sequence including heavily swaying trees [30], as is shown in Fig. 4. To do this, we first extract the descriptive NIPE vectors $V_t^p$ of image patch $P_t$ at each frame, and check how these NIPE vectors change over a period of time. If the image patch $P$ is not occupied by a foreground object within a time interval, these descriptive NIPE vectors $V_t^p$ should be stable and obey the underlying background model. Otherwise, if the image patch $P_t$ is occupied by a foreground object, the corresponding extracted NIPE vector would disobey the underlying background model. In this work, we choose LBP histogram intersection as a robust similarity measure between patches. Then, the low dimensional NIPE vector is computed as a vector whose components are the normalized LBP histogram intersections between current image patch and its neighboring image patches. The basic $3 \times 3$ LBP operator is adopted in our implementation. The advantages of using the basic $3 \times 3$ LBP operator are as follows. First, LBP is a powerful means of texture description. Second, LBP is invariant with respect to monotonic gray-scale changes. This is very valuable when the scenes are under changing illumination conditions. Third, LBP is very fast to compute, which is an important property for the real-time applications.

Fig. 5 plots the evolving curves (820 frames) of the distance between the current NIPE vector and the background NIPE vector (obtained by the modeling procedure described in subsection III.C), for the three patches separately. In our paper, we use L1 distance (see Eq.(2)). It is obvious to see that distance distribution of Patch A is fairly stable near zero, in consistent with the fact that Patch A is a flat region in the sky. For the dynamic Patch B, the fluctuation of the corresponding distance is relatively small. When no foreground object occupies the Patch C before frame 392, the distribution of the corresponding distance is also relatively stable and low. However, when a foreground object
3.3. Background modeling based on NIPE

In this section, we introduce background modeling mechanism based on NIPE described above. The goal is to maintain and construct a statistical representation of the scene that the camera sees. We partition each new video frame into a set of non-overlapping patches with $N \times N$ pixels and represent each patch by a $K$ dimensional NIPE vector. In the following, we explain the background modeling procedure for each image patch.

We consider the NIPE vectors of a particular patch over time as a patch process, and model the background model for this patch as a group of weighted adaptive NIPE vectors, $\{H_1, H_2, ..., H_n\}$, where $n$ is the number of model NIPE vectors. Each model NIPE vector has a weight between 0 and 1 and all the weights sum up to 1. The weight of the $i$th model NIPE vector is denoted by $w_i$. At the beginning phase of detection, we sort the model NIPE vectors in decreasing order according to their weights and the first $B$ model NIPE vectors are chosen as the background model:

$$B = \arg\min_b \left( \sum_{i=1}^{b} w_i > T_B \right)$$ (1)

where $T_B$ is a measure of the minimum portion of the data that should be accounted for by the background. This takes the "best" NIPE vectors until a certain portion $T_B$ of the recent data has been accounted for. An incoming NIPE vector $V$ of the given patch is checked against the existing $n$ model NIPE vectors until a match is found. In our paper, the distance between two NIPE vectors $V_1$ and $V_2$ is calculated by the following distance rule:

$$L(V_1, V_2) = \sum_{i=1}^{K} |V_1 - V_2|$$ (2)

where $K$ is the number of neighboring patches and $L(V_1, V_2)$ is the L1 distance. Note that the smaller the distance, the higher the probability of matching. If the distance is smaller than the threshold $T_s$ for at least one background model, the patch is classified as background. Otherwise, the patch is labeled as foreground.

In the background updating phase, if none of the $n$ model NIPE vectors match the current NIPE vector $V$, the model NIPE vector with lowest weight is replaced with the current NIPE vector $V$ and a low prior weight $\beta$. In our experiments, a value of $\beta=0.05$ is used and a match is defined as the distance below a threshold $T_s$. The weights of the $n$ model NIPE vectors at time $t+1$ are adjusted with the new data as follows:

$$w_{i,t+1} = (1-\gamma)w_{i,t} + 2M_{i,t+1}$$ (3)

where $\gamma$ is the learning rate and $M_{i,t+1}$ is 1 for the model which matched and 0 for the remaining models. After this approximation, the weights are renormalized. The bigger the weight, the higher the probability of being a background NIPE vector. The adaptation speed of the background model is controlled by the learning rate parameter $\gamma$. The bigger the learning rate, the faster the adaptation is. Here, we use an adaptive learning rate $\gamma$, the value of which is inversely proportional to the distance measure between two NIPE vectors. The unmatched model NIPE vectors remain the same. The model NIPE vector which matches the new observation is updated as follows:

$$H_{i,t+1} = (1-\gamma)H_{i,t} + \gamma V$$ (4)

4. Mixture of Gaussian model (GMM) for BGS

In this paper, we incorporate the popular GMM [2] method with our NIPE-based BGS method [19]. The GMM method can deal with periodic motions from a cluttered background, slow lighting changes, etc. However, it cannot adapt to the quick variations in dynamic environments, such as tree leaves swaying, water rippling, and camera jitter. In other words, the GMM method generates large number of false foreground pixels under those difficulty conditions (please see Figs. 6–9).

In the GMM, three significant parameters $K_{gmm}$, $T_{gmm}$ and $x_{gmm}$ are needed to be set, where $K_{gmm}$ is the number of Gaussian components, $T_{gmm}$ the minimum portion of the background model and $x_{gmm}$ the learning rate. In our implementation, we set $K_{gmm}=3$ (three Gaussians), $T_{gmm}=0.7$ and $x_{gmm}=0.01$. Please refer to [2] for more details about the GMM method.

5. Heuristic seeds selection for matting

The seed regions play an important role in the alpha matting process. Previous matting methods involve a tremendous amount of manual interaction to provide the seed regions. Since the object of interest can be accurately localized and the small movements of the dynamic scenes are eliminated after the background subtraction and connected components extraction stage, we can adopt the heuristic seeds selection scheme described in Section 2 to automate the matting process. More specifically, as shown in Fig. 2, we first use a connected components algorithm to give a green bounding box of the labeled foreground pixels. Then, a yellow bounding box is obtained by doubling the size of the green bounding box, and a red bounding box is obtained by extending the size of the green bounding box in each dimension a quarter. In the image obtained by fusing the NIPE-based method and the GMM-based method, the white pixels included in the red bounding boxes are foreground seeds. Meanwhile, the black pixels outside the red bounding boxes and inside the yellow bounding boxes are background seeds. According to these foreground and background seeds, our goal is to classify the black pixels within the red bounding boxes as foreground or background using the popular matting method. In this paper, we choose the closed form matting algorithm [31] to refine the detection results.

To extract a refined detection map $\gamma$ matching the seeds obtained by our heuristic seeds selection scheme, we rewrite $\gamma$ in its vector form and minimize the following cost function

$$E = \gamma^T L \gamma + \sum_{i=1}^{n} \frac{1}{w_k} \left( 1 + \frac{e_k}{w_k} \right) \left( \mu_i - \mu_k \right)^T \left( \mu_i - \mu_k \right)$$ (5)

where $L$ is the Matting Laplacian matrix proposed by Levin [31], and $S$ is the set of selected seeds, in which $s_i=0$ and $s_j=1$ indi cate background seed and foreground seed respectively. The $(ij)$ element of the matrix $L$ is defined as:

$$L_{ij} = \frac{\delta_{ij} - \frac{1}{w_k} \left( 1 + \frac{e_k}{w_k} \right) \left( \mu_i - \mu_k \right)^T \left( \mu_i - \mu_k \right)}{\sum_k \frac{1}{w_k} \left( \mu_i - \mu_k \right)^T \left( \mu_i - \mu_k \right)}$$ (6)

where $\delta_{ij}$ and $L$ are the colors of the input image $l$ at pixels $i$ and $j$, $\delta_{ij}$ is the Kronecker delta, $\mu_k$ and $\sum_k$ are the mean and covariance matrix of the colors in window $w_k$, $L$ is a $3 \times 3$ identity matrix, $e$ is a regularizing parameter, and $|w_k|$ is the number of pixels in the window $w_k$.

6. Experiments

In this section, we apply our method to several challenging video sequences, and systematically compare it with several
representatives of state-of-the-art in BGS. Moreover, some specific properties of our method are empirically highlighted by a group of carefully designed experiments. Before indulging in the evaluation and analysis, we first introduce the setting of our experiment.

6.1. Experiment setting

In our experiments, our method is implemented using C++, on a computer with an Intel-Core 2 1.86 GHz processor and 2 GB of memory. Typically, the number of steps to converge the
optimization of Eq. (5) is less than 50 and the processing time on our computer is about 0.02 s using the data obtained by our heuristic selection scheme. Our method has achieved the processing speed of 5 fps at the resolution of 320 × 240 pixels. To the best of authors’ knowledge, there is still a lack of globally accepted baseline algorithms for the extensive evaluation of BGS algorithms. Few of BGS algorithms are open source. Even that we can implement those methods, the parameters tuning is always a problem to achieve the results reported in original literature. This makes the comparison of the different approaches rather difficult. For performance evaluation, we compare our approach against several representatives of the current state-of-the-art in
FN and FP stand for false negatives and false positives, respectively.

Both qualitative and quantitative comparisons are used to evaluate our method. The qualitative comparison is done in terms of the matting results and the segmentation results. In our method, a pixel is classified as foreground if its alpha matte $\gamma$ obtained from the matting process is larger than 0.5. Otherwise, it is detected as background. This threshold can be perturbed with little effect on performance. Moreover, the quantitative comparison is done in terms of the number of false negatives (the number of foreground pixels that are missed) and false positives (the number of background pixels that are marked as foreground). Identical parameters are used in the test sequences (please see Figs.6–9), although better results could be obtained by customizing the values for each sequence. Specifically, in our NIPE-based BGS method, we set $K=8$, $N \times N = 12 \times 12$, $n = 3$, $T_p = 0.7$, $T_s = 1$, $\alpha = 0.01$ and $\beta = 0.05$.

We also have compared the speeds of our method, GMM, KDE and CodeBook. For the experiment setting used in the tests, the frame rates of 5, 15, 10 and 10 fps were achieved by our method, GMM, KDE and CodeBook, respectively.

6.2. Comparison with other methods

The qualitative comparison results of our method, the GMM, the KDE and the CodeBook on the used test sequences are shown in Figs. 6–9, respectively. In these figures, the first row contains the original video frames. The second row, third row and fourth row contain the detection results of the GMM, KDE and CodeBook, respectively. The last two rows contain the matting and segmentation results of our method, respectively.

Specifically, Fig. 6 shows qualitative comparison results on the Marine_Aquarium sequence from the software of Marine Aquarium 2. The Marine_Aquarium sequence contains bubbles and illumination changes, with foreground (i.e., fishes) composed of complicated shapes. This is a very difficult scene from the moving object segmentation and matting point of view. Our method gives good results because it exploits the complementary strengths of the GMM-based method and the NIPE-based method to provide the seeds for accurate moving object matting and segmentation. While the GMM generates large number of false foreground pixels under this difficulty condition, due to the non-periodic motions of bubbles and the quick variations of illumination. The KDE allows for better resilience against dynamic backgrounds due to pixel values that could be explained away by distributions of neighboring pixels are reclassified as background in the second phase of the approach. Since sample background values at each pixel are quantized into codebooks to represent a compressed form of background model in the CodeBook, the results obtained by the CodeBook have greatly improved. However, there are still some false foreground pixels under this difficulty condition, due to the quick variations and the non-periodic motions of moving backgrounds and illumination.

Fig. 7 shows qualitative comparison results on the Waving_Trees sequence from [30]. The challenge from the Waving_Trees sequence is the heavily swaying trees. This causes classical BGS methods (e.g., GMM) that rely only on the pixel color or intensity information to make the decision to fail. The KDE achieves better resilience against dynamic backgrounds (e.g., the third row of Fig. 6) due to pixel values that could be explained away by distributions of neighboring pixels are reclassified as background in the second phase of the approach. However, there are still some problems which lead to poor performance when infrequent motions occur, such as trees rustling periodically (but not constantly) due to wind gusts in the third row of Fig. 7. It is obvious to see that, for the CodeBook, some false positives occur on the areas of the moving trees (see the fourth row of Fig. 7). This is because the background in the CodeBook is encoded on a pixel-by-pixel basis. Since we explicitly designed a novel BGS method as attention mechanisms to generate many possible foreground pixels by tuning it for low false-positives and false-negatives as much as possible, our method manages the situation relatively well.

Fig. 8 shows qualitative comparison results on the Bottle sequence. The Bottle sequence contains a moving bottle in foreground, with dynamic background composed of ripples in the water. The GMM generates some false foreground pixels due to the non-periodic motions of the ripples in the water. The KDE and the CodeBook generate some false background pixels on image areas where the color of foreground and background is similar. It can be seen that our method has almost no false detections in spite of the dynamic motions.

Fig. 9 shows qualitative comparison results on the Camera_Jitter sequence from [8]. The Camera_Jitter sequence contains average camera jitter of about 14.06 pixels. Since the nominal motion of the camera do not repeat exactly, the GMM handles this difficulty condition poorly, i.e., producing many false foreground pixels. While the KDE and our method manage the situation relatively well, due to considering the meaningful correlation between pixels in the spatial vicinity. Moreover, the CodeBook can also handle the situation relatively well due to a compact background model represented by quantized codewords.

In order to provide a quantitative perspective about the quality of foreground detection with our method, we manually mark the foreground regions in all frames from each sequence to generate ground truth data, and make comparison with the GMM, the KDE and the CodeBook. We sum the error from the frames corresponding to the ground truth frames. The numbers of error classifications are achieved by the average error per frame. The corresponding quantitative comparison is reported in Fig. 10. For most of the test sequences, our method achieves best performance in terms of the sum of false positives and false negatives. According to the overall results, our method can automatically pull a high-quality alpha matte and foreground from the dynamic scenes.

6.3. Efficacy of background subtraction driven seeds selection

One big advantage of our algorithm lies in that we can fuse the NIPE-based method using block feature and the GMM-based method using pixel feature to automatically provide the
foreground seeds for the matting algorithms. To verify this advantage, we check the foreground seeds provided for the matting algorithms over time in all the test sequences and give several typical examples in Fig. 11. The first column contains the original video frames. The second and third columns contain the detection results of GMM and the NIPE respectively. The fourth column contains the foreground seeds obtained by combining the NIPE-based method and the GMM-based method. In the fourth column, the white pixels included in the red bounding boxes are the foreground seeds. Meanwhile, the black pixels outside the red bounding boxes and inside the yellow bounding boxes are the background seeds. According to these foreground and background seeds, our goal is to classify the black pixels within the red bounding boxes as foreground or background using the proposed matting method. The last column contains the detection results of our method. As expected, our background subtraction driven seeds selection strategy can automatically provide appropriate foreground seeds for matting algorithm. Moreover, it is obvious to see from Fig. 11 that our method outperform the GMM, the NIPE and the combination of both for the used testing sequences with dynamic backgrounds.

7. Conclusion

We address in this paper the automatic moving object segmentation and matting in dynamic scenes and have introduced a new BGS driven seeds selection method for computing the solutions by simultaneously utilizing matting techniques. The sequences we consider contain bubbles, water ripples, swaying trees and camera jitter. Three main steps compose our method. First, a novel BGS method, in which advantages of two complementary BGS methods (i.e., the NIPE and the GMM) are effectively combined, is used as attention mechanisms for generating many possible foreground pixels by tuning it for low false-positives and false-negatives as much as possible. The bounding boxes of the labeled foreground pixels are then extracted using a connected components algorithm. Finally, matting of the object associated to a given bounding box is performed using a heuristic seeds selection scheme, in which the labeled foreground pixels in the bounding box are used as the foreground seeds and the pixels in a window of slightly larger extent than the bounding box (that) are used as the background seeds. Experiments and comparisons to other moving object detection methods on challenging sequences demonstrate the performance of our method for video analysis in dynamic scenes.

Acknowledgment

We thank Prof. Ahmed Elgammal for kindly sharing his code [7] and Dr. Kim for sharing his binary code. This work is supported by Natural Science Foundation of China (No.61202299, No.61170179 and No.61003115), China Postdoctoral Science Foundation (No.2011M501081), Scientific Research Foundation of Huaqiao University (No.11BS109), Fundamental Research Funds for the Central Universities (No.JB-ZR1219 and No.JB-ZR1217), and Science and Technology Planning Project of Fujian Province, China (No.2011H0016 and No.2011H0028).

References

Bineng Zhong received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2004, 2006, and 2010, respectively. From 2007 to 2008, he was a Research Fellow with the Institute of Automation and Institute of Computing Technology, Chinese Academy of Science. Currently, he is a Lecturer with the School of Computer Science and Technology, Huaqiao University, Xiamen, China, and he is also a Post-Doc with the School of Information Science and Technology, Huaqiao University, Xiamen, China. His current research interests include pattern recognition, machine learning, and computer vision.

Yan Chen received the B.S. and Ph.D. degrees in computer science from University of Science and Technology of China, Hefei, China, in 2003 and 2010, respectively. Currently, she is a Lecturer with the School of Computer Science and Technology, Huaqiao University, Xiamen, China. Her current research interests include computer vision, image retrieval, and pattern recognition.

Yewang Chen received the B.S. and M.S. degrees in information management and computer science from Huaqiao University, Xiamen, China, in 2001 and 2006, respectively, and the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2009. Currently, he is a Lecturer with the School of Computer Science and Technology, Huaqiao University, Xiamen, China. His current research interests include natural language processing, ontology and pattern recognition.

Rongrong Ji received the B.S. degree in electronic engineering from Harbin Engineering University, Harbin, China, in 2005, and the Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, China, in 2011. Currently, he is a Post-Doc with the Columbia University, USA. His current research interests include computer vision image retrieval, and pattern recognition.

Ying Chen received the B.S. degree in department of mathematics from Beijing University in 2000, and Ph.D. degree in applied mathematics from Academy of Mathematics and Systems Science, Chinese Academy of Sciences in 2005. He works in Beijing Electronic Sciences and Technology Institute of China and Beijing University, Xiamen, China. His current research interests include natural language processing, ontology and pattern recognition.
Duansheng Chen received the B.S. degree in remote sensing from Zhejiang University, Hangzhou, China, in 1982, the M.S. degree in remote sensing from Nanjing Forestry University, Nanjing, China, in 1986, and the Ph.D. degree in information and communication from University of Science and Technology of China, Hefei, China, in 2005. Currently, he is a Professor with the School of Computer Science and Technology, Huazhao University, Xiamen, China. His current research interests include pattern recognition, machine learning, and computer vision.

Hanzi Wang is currently a “Minjiang” Distinguished Professor at Xiamen University, China and an Adjunct Professor at the University of Adelaide, Australia. He was a Senior Research Fellow (2008–2010) working with Prof. David Suter, at the University of Adelaide, Australia; an Assistant Research Scientist (2007–2008) and a Postdoctoral Fellow (2006–2007) working with Prof. Gregory D. Hager, at the Johns Hopkins University; and a Research Fellow working with Prof. David Suter, at Monash University, Australia (2004–2006). He received the Ph.D. degree in Computer Vision from Monash University. He was awarded the Douglas Lampard Electrical Engineering Research Prize and Medal for the best Ph.D. thesis in the Department. His research interests are concentrated on computer vision and pattern recognition, including visual tracking, robust statistics, object detection, video segmentation, model fitting, optical flow calculation, 3D structure from motion, image segmentation, and related fields. He is a Senior Member of the IEEE. He is an Associate Editor for IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT) and he was a Guest Editor of Pattern Recognition Letters (September 2009).